



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/645,819	08/20/2003	Robin J. Green	SONYP024	8040

25920 7590 09/06/2006

MARTINE PENILLA & GENCARELLA, LLP  
710 LAKEWAY DRIVE  
SUITE 200  
SUNNYVALE, CA 94085

EXAMINER

REPKO, JASON MICHAEL

ART UNIT	PAPER NUMBER
2628	

DATE MAILED: 09/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b> 10/645,819	<b>Applicant(s)</b> GREEN ET AL.	
	<b>Examiner</b> Jason M. Repko	<b>Art Unit</b> 2628	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☐ Responsive to communication(s) filed on \_\_\_\_.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-46 is/are pending in the application.  
     4a) Of the above claim(s) \_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-46 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 29 December 2005 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
     a) ☐ All    b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |   |  |
|---|--|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. ____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                  | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)            |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date ____ | 6) <input type="checkbox"/> Other: ____  |

## **DETAILED ACTION**

### ***Response to Arguments***

1. Applicant's arguments filed 7/31/2006 have been fully considered but they are not persuasive.
2. With regard to the rejection of claims 6-11 and 17-22 under 35 USC 112 first paragraph, Applicant argues that the subject matter is fully supported by Figures 14 and 15 and the text on pages 25-28 of the application. In the Office action dated July 24, 2006, it was asserted the language "generating a ray from a point on the object without accessing pre-calculated geometry associated with the object" added by amendment to claims 6 and 17 is not supported by the descriptive portion of the specification. Figures 14 and 15 show a point "p" on the surface of an object as the origin of a ray 298. Figures 14 and 15 do not provide support for the claim language "without accessing pre-calculated geometry associated with the object." In fact, the "precomputed geometry" of the object 282 is illustrated in the figure. The corresponding detailed description in the specification for Figure 14 (pages 25-28 as directed by Applicant's remarks) describes checking ray intersections with object geometry 282 (lines 21 on page 25 through line 3 on page 26: "Next, the segment within voxel C 292 is examined and it is determined that at point 302 ray 298 intersects with a surface of object 282"), and testing polygons (lines 19-20 on page 26: "Here again, the desired result has been obtained when only testing a minimal amount of polygons, and it is possible to move to the next ray."): The corresponding detailed description in the specification for Figure 15 describes using previously generated ray 298 computed at point P 270 (line 12 on page 28), where point P was on the precomputed object geometry. The step of "generating a ray associated with a point on the display object" appears in paragraphs 12, 13, 14,

15, 16, 53 (line 6-7, p. 16) and 67 in describing a ray tracing algorithm, but these paragraphs do not disclose how to generate a ray from a point on the object without accessing the pre-computed geometry with the object. There is no support for the claimed limitation in Figures 14 and 15, or in the corresponding detailed description as alleged by Applicant, as the only mentioned relationship between the generated ray and the precomputed geometry is generating the ray based on the geometry and testing the ray against the geometry.

3. In response to applicant's argument one of ordinary skill in the art would not have combined Sloan and Purcell to arrive at the claimed invention, Applicant's attention is directed to the first paragraph of section 5. Specifically, Sloan et al discloses in the first paragraph of section 5: "As a preprocess, we perform a global illumination simulation over an object **O** using the SH basis over the infinite sphere as emitters." Sloan et al then states that the "light gathering solution technique is a straightforward adaptation of existing approaches [7][25] and could be accelerated in many ways," noting that [7] is a ray-tracing algorithm." Sloan et al is able to provide "real-time rendering in dynamic, low-frequency lighting environments," because the results of the global illumination preprocess step are used. As shown in the prior art rejection, Purcell provides a system and method to accelerate the preprocessing step. The advantages and applicability of the Purcell system to the method disclosed by Sloan et al would have been readily apparent to one of ordinary skill in the art.

4. In response to applicant's remarks on page 16 that the back and forth processing would be prohibitively expensive, it is noted that the preamble of claim 1, for example, states, "presenting lighting characteristics associated with a display object in real time." As stated in the title of the reference, the Sloan et al reference discloses presenting in real-time. Admittedly, Sloan et al does

Art Unit: 2628

not disclose performing the preprocessing step in real time. As previously stated, Purcell et al teaches an accelerated version of ray-tracing. “Presenting” and “rendering” an object in real-time does not necessarily mean performing frame by frame every graphics processing operation that contributes to the presented image. Since feature upon which applicant relies are not recited in the rejected claim, the Applicant’s argument is moot. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

5. With regard to the language added to the preamble of the claims 1, 6, 12, 17, 23, 28, 33, and 36 (“through a single graphics processing unit”), Sloan et al admits that due to the limitations of contemporary consumer-level hardware (e.g. PIII-933 PC with an ATI Radeon 8500) the dot product of the captured samples and the basis functions must be performed in software. However, Sloan et al teaches, “ideally, this computation would be performed on graphics hardware” (3<sup>rd</sup> paragraph of section 6.2) and Sloan et al states, “we are interested in tracking fast-improving PC graphics hardware so that all computation, including transfer matrix transforms and SH-rotation, may eventually be performed on the GPU” (2<sup>nd</sup> paragraph of section 10). Therefore, at the time of the invention, it would have been obvious to a person of ordinary skill in the art to perform all computation on the GPU as suggested by Sloan et al. The motivation for doing so would have been to improve the performance of the method. Therefore, it would have been obvious to further modify the combination of Sloan et al and Purcell et al to obtain the invention specified in the claims.

***Claim Rejections - 35 USC § 112***

6. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

7. **Claim 6-11 and 17-22 rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement.** The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

8. With regard to the language “generating a ray from a point on the object without accessing pre-calculated geometry associated with the object” in claims 6 and 17, the step of “generating a ray associated with a point on the display object” appears in paragraphs 12, 13, 14, 15, 16, 53 (line 6-7, p. 16) and 67 in describing a ray tracing algorithm, but these paragraphs do not disclose how to generate a ray from a point on the object without accessing the pre-computed geometry with the object. With respect to “determining secondary illumination features,” the Applicant discloses that the ray tracing calculations are first computed to determine “the geometry associated with the object and the object’s environment” (lines 10-11, p. 30) and secondary lighting contribution is determined using the data resulting from the ray tracing, but does not describe “generating a ray from a point on the object without accessing pre-calculated geometry associated with the object.” Furthermore, paragraphs 72-76, figures 14, 15, 17, and 18 do not convey to one skilled in the relevant art how generating rays can be accomplished in this manner.

***Claim Rejections - 35 USC § 101***

9. 35 U.S.C. 101 reads as follows:

Art Unit: 2628

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

10. Claims 1-46 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

11. Claims 1-11 and 23-32 appear to be to an abstract idea rather than a practical application of the idea. Claims 1-11 and 23-32 does not result in a physical transformation nor does it appear to provide a useful, concrete and tangible result. Specifically, it does not appear to produce a tangible result because merely determining lighting characteristics are nothing more than thoughts or computations within a processor. It fails to use or make available for use the result of the computations to enable its functionality and usefulness to be realized. Additionally, the asserted practical application in the specification of the computations is displaying the resulting computer graphics image on a display screen. The practical application is not explicitly recited in the claims nor does it flow inherently therefrom.

12. Claims 12-22 and 33-38 are directed to a “computer readable medium,” and in paragraph 85 of the descriptive portion of the specification, a computer readable medium is defined as including “electromagnetic wave carriers.” A wave is nothing but the physical characteristics of a form of energy, and as such is nonstatutory natural phenomena.

13. Claims 39-46 are directed to a computer that solely calculates a mathematical algorithm which is non-statutory subject matter. Claims 39-46 are directed to a generic computing system (i.e. video game console, display screen in communication with a GPU) performing a mathematical algorithm without making available the results of the computation. In effect, claims 39-46 seek to cover every substantial practical application of the abstract idea itself.

Art Unit: 2628

14. To expedite a complete examination of the instant application, the claims rejected under 35 U.S.C. 101 as non-statutory subject matter are further rejected as set forth below in anticipation of applicant amending the claims to place them within the four categories of invention.

***Claim Rejections - 35 USC § 102***

15. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

**16. Claims 39, 42 and 46 are rejected under 35 U.S.C. 102(b) as being anticipated by Jan Kautz, Peter-Pike Sloan and John Snyder, “Fast, Arbitrary BRDF Shading for Low-Frequency Lighting Using Spherical Harmonics,” June 26, 2002, Proceedings of the 13th Eurographics Workshop on Rendering, p. 291-296, 335 (Kautz et al).**

17. With regard to claim 39, Kautz et al discloses “a computing device, comprising:

- a. a graphics processing unit (GPU) capable of determining lighting characteristics for an object in real time without accessing pre-calculated geometry associated with the object (*3<sup>rd</sup> paragraph of section 4: "The remaining operations (steps 2-4) are performed on the GPU."; steps 2-4 in the 1<sup>st</sup> paragraph of section 4, p. 293; section 6: "...for the fixed view/fixed light modes we achieve real-time frame rates (Table 1)."*),
- b. the lighting characteristics defined through a basis function (*2<sup>nd</sup> paragraph of section 3: "We first parameterize  $f^*$  by local view direction to get spherical functions, which we represent in the SH basis via..."*),



Art Unit: 2628

- c. the GPU including a stream processor configured to split a stream of data associated with the lighting characteristics into multiple simultaneous operations (*3<sup>rd</sup> paragraph of section 4: "View vector rotation is done in a vertex shader, the BRDF lookup is just a texture access, and the final dot product is computed in a pixel shader."*),
- d. the lighting characteristics further determined without preprocessing data to determine a fixed transfer function" (*9<sup>th</sup> paragraph of section 2: "Alternatively, the methods of this paper can be used without precomputed transfer, so that no storage or preprocessing is needed for transfer vectors or matrices over an object."*).
18. With regard to claim 42, Kautz et al discloses "wherein the stream processor is a programmable hardware unit capable of executing code that is replicated multiple times" (*3<sup>rd</sup> paragraph of section 4: "View vector rotation is done in a vertex shader, the BRDF lookup is just a texture access, and the final dot product is computed in a pixel shader."*).
19. With regard to claim 46, Kautz et al discloses "the basis function is one of a wavelet and a spherical basis function" (*2<sup>nd</sup> paragraph of section 3: "We first parameterize  $f^*$  by local view direction to get spherical functions, which we represent in the SH basis via..."*).

***Claim Rejections - 35 USC § 103***

20. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2628

21. The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

22. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

23. **Claims 1-5, 12-15 and 28-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peter-Pike Sloan, Jan Kautz, John Snyder, "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments," July 2002, ACM Transactions on Graphics, Vol. 21 No. 3, p. 527-536 (Sloan et al) in view of Timothy J. Purcell, Ian Buck, William R. Mark, Pat Hanrahan, "Ray Tracing on Programmable Graphics Hardware," July 2002, ACM Transactions on Graphics, Vol. 21, No. 3, p. 703-712 (Purcell et al) in further view of U.S. Patent Application Publication No. 2001/0028352 to Neagle et al.**

24. With regard to **claim 1**, Sloan et al teaches “a method for presenting lighting characteristics associated with a display object in real-time, comprising: executing a ray tracing algorithm that includes generating a ray associated with a point on the display object” (5<sup>th</sup> paragraph of section 5: “In the first pass, for each  $p \in O$ , we cast shadow rays in the hemisphere about  $p$ ’s normal  $N_p$ , using the hierarchy to cull directions outside the hemisphere.”). Sloan et al teaches “determining an approximation of a transfer function component using at least one basis function” in the equations given in the 6<sup>th</sup> and 7<sup>th</sup> paragraphs of section 5. Sloan et al does not use this explicit language; however one of ordinary skill in the art would recognize this feature from the description of the terms of the equation in the 2<sup>nd</sup> paragraph of section 5 and the 3<sup>rd</sup> paragraph of section 4:

*A transfer matrix  $(M_p)_{ij}$  is useful for glossy surfaces and represents a linear transformation on the lighting vector which produces projection coefficients for an entire spherical function of transferred radiance  $L'_p(s)$  rather than a scalar...Components of  $(M_p)_{ij}$  represent the linear influence of the  $j$ -th lighting coefficient of incident radiance  $(L_p)_j$  to the  $i$ -th coefficient of transferred radiance  $(L'_p)_i$ .*

25. Sloan et al does not teach ray tracing on stream processor. Purcell et al teaches “executing a ray tracing algorithm through a stream processor” (Figure 2, page 705).

26. With regard to **claim 28**, Sloan et al teaches “calculating a lighting function for an object to be rendered using a basis function, comprising: calculating a transfer function approximation of the lighting function,” (3<sup>rd</sup> paragraph of section 4: “A transfer matrix  $(M_p)_{ij}$  is useful for glossy surfaces and represents a linear transformation on the lighting vector which produces projection coefficients for an entire spherical function of transferred radiance  $L'_p(s)$  rather than

Art Unit: 2628

*a scale.”) “the lighting function being sampled at a texel” (3<sup>rd</sup> paragraph of section 6: “The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader.”; 5<sup>th</sup> paragraph of section 6.2: “The basis function textures are also supersampled and decimated in the same way as a preprocess.”). Furthermore, Sloan et al states in the 4<sup>th</sup> paragraph of section 1:*

*The object’s shaded “response” to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral.*

27. Sloan et al does not teach using a stream processor to compute the rays, which determines the transfer function (*1st paragraph of section 5: “As a preprocess, we perform a global illumination simulation over an object  $O$  using the SH basis over the infinite sphere as emitters.”*). Purcell et al teaches “executing a ray tracing algorithm through a stream processor” (*Figure 2, page 705*).

28. With regard to **claims 1 and 28**, Sloan et al states that the “light gathering solution technique is a straightforward adaptation of existing approaches [7][25] and could be accelerated in many ways,” noting that [7] is a ray-tracing algorithm. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to accelerate the computation of radiance transfer as taught by Sloan et al, by using a stream processor to trace rays as taught by Purcell et al. The motivation for doing so would have been to achieve better performance tracing rays as stated by Purcell et al in section 2.3, which would be advantageous for interactive systems.

Art Unit: 2628

29. With regard to **claims 1 and 28**, Sloan et al discloses sampling a texel associated with a corresponding point on an object (*1<sup>st</sup> paragraph of section 6: "We now have a model O capturing radiance transfer at many points p over its surface, represented as vectors or matrices."*; *3<sup>rd</sup> paragraph of section 6: "The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader."*; *5<sup>th</sup> paragraph of section 6.2: "The basis function textures are also supersampled and decimated in the same way as a preprocess."*), but does not disclose a sampling location within the texel. Neagle et al discloses sampling at the "center point of a pixel" (*paragraph 256: "In one embodiment, the graphics system ensures that one of the rendered samples lies in the center of the bin or pixel area."*).

30. With regard to **claims 1 and 28**, at the time of the invention, it would have been obvious to a person of ordinary skill in the art to use the sampling scheme disclosed by Neagle et al to sample the texels in the system and method disclosed by Sloan et al. The motivation for doing so would have been to sample at a location that is more representative of the given texel than any neighboring texel, an advantage well-known in the art.

31. With regard to **claims 1 and 28**, Sloan et al admits that due to the limitations of contemporary consumer-level hardware (e.g. PIII-933 PC with an ATI Radeon 8500) the dot product of the captured samples and the basis functions must be performed in software. However, Sloan et al teaches, "ideally, this computation would be performed on graphics hardware" (*3<sup>rd</sup> paragraph of section 6.2*) and Sloan et al states, "we are interested in tracking fast-improving PC graphics hardware so that all computation, including transfer matrix transforms and SH-rotation, may eventually be performed on the GPU" (*2<sup>nd</sup> paragraph of section 10*). Therefore, at the time of the invention, it would have been obvious to a person of ordinary skill

in the art to perform all computation on the GPU as suggested by Sloan et al. The motivation for doing so would have been to improve the performance of the method. Therefore, it would have been obvious to further modify the combination of Sloan et al, Purcell et al and Neagle et al to obtain the invention as specified in **claims 1 and 28**.

32. **Claim 2** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches “determining whether the ray is within a view plane of a light source” (5<sup>th</sup> paragraph of section 5: “In the first pass, for each  $p \in O$ , we cast shadow rays in the hemisphere about  $p$ ’s normal  $N_p$ ... We tag each direction  $s_d$  with an occlusion bit,  $1 - V_p(s_d)$ , indicating whether  $s_d$  is in the hemisphere and intersects  $O$  again (i.e., is self-shadowed by  $O$ ).”). Sloan et al does not use this explicit language in the fifth paragraph of section 5; however, one of ordinary skill in the art would recognize that a ray occluded by an object is not in the view plane of the light.

33. **Claim 3** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches “if the ray is within the view plane of the light source, then the method includes, determining a direct illumination component of the lighting characteristic” (5<sup>th</sup> paragraph of section 5: “completely unoccluded bins/samples receive only direct light from the environment.”).

34. **Claim 4** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches “if the ray is not within the view plane of the light source, then the method includes, determining a self interreflection component of the lighting characteristic” (5<sup>th</sup> paragraph of section 5: “Self-occluded directions and bins are tagged so that we can perform further interreflection passes on them”).

Art Unit: 2628

35. **Claim 5** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches “repeating the determining of an approximation of a transfer function component for a series of basis functions without accessing a pre-calculated geometry associated with the object” (2<sup>nd</sup> paragraph of section 4: “In other words, each component of  $(M_p)_i$  represents the linear influence that a lighting basis function  $(L_p)_i$  has on shading at  $p$ .”; 2<sup>nd</sup> paragraph of section 6.2: “For efficiency, we precompute textures for the basis functions weighted by differential solid angle...each evaluated over the cube map parameterization for  $s$ . The resulting integral then becomes a simple dot product of the captured samples of  $L_p(s)$  with the textures  $B^m_l(s)$ .”; 3<sup>rd</sup> paragraph of section 6: “The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader.”); and rendering the display object using the approximation of the transfer function component for the series of basis functions” (1<sup>st</sup> paragraph of section 6: “Rendering [model]  $O$  requires the following steps at run-time: ...compute incident lighting  $\{L_{p_i}\}$  at one or more sample points  $P_i$  near  $O$  in terms of the SH basis.”).

36. **Claims 12-15** are rejected with the rationale of claims 1-4, respectively. Claims 12-15 are similar in scope to claims 1-4 recited as a computer readable medium having program instructions (shown in section 9 of Sloan et al).

37. **Claim 29** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches a “transfer function approximation is associated with the basis function that characterizes a global illumination associated with the object” (2<sup>nd</sup> paragraph of section 4: “A transfer vector  $(M_p)_i$  is useful for diffuse surfaces and represents a linear transformation on the

*lighting vector producing scalar exit radiance...each component of  $(M_p)_i$  represents the linear influence that a lighting basis function  $(L_p)_i$  has on shading at  $p$ . ”).*

38. **Claim 30** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches a “transfer function approximation is a set of coefficients configured to describe a surface reflectance” (*4<sup>th</sup> paragraph of section 4: “Components of  $(M_p)_{ij}$  represent the linear influence of the  $j$ -th lighting coefficient of incident radiance  $(L_p)_j$  to the  $i$ -th coefficient of transferred radiance  $(L'_p)_i$ ”*).

39. **Claim 31** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches rendering an object (*1<sup>st</sup> paragraph of section 6: “We now have a model  $O$  capturing radiance transfer at many points  $p$  over its surface, represented as vectors or matrices. Rendering  $O$  requires the following steps at run-time: ... ”*).

40. **Claims 6-11, 17-24, 26, 33, 34, and 36-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peter-Pike Sloan, Jan Kautz, John Snyder, "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments," July 2002, ACM Transactions on Graphics, Vol. 21 No. 3, p. 527-536 (Sloan et al) in view of Timothy J. Purcell, Ian Buck, William R. Mark, Pat Hanrahan, “Ray Tracing on Programmable Graphics Hardware,” July 2002, ACM Transactions on Graphics, Vol. 21, No. 3, p. 703-712 (Purcell et al).**

41. With regard to **claim 6**, Sloan et al teaches "a method for determining secondary illumination features for an object to be displayed, comprising: calculating an approximation to a transfer function associated with at least one basis function” in the equations given in the 6<sup>th</sup> and 7<sup>th</sup> paragraphs of section 5 (as shown with regard to claim 1); “wherein the approximation to the



Art Unit: 2628

transfer function represents a component of the secondary illumination features" (4<sup>th</sup> paragraph of section 5: "Components of  $(M_p)_{ij}$  represent the linear influence of the  $j$ -th lighting coefficient of incident radiance  $(L_p)_j$  to the  $i$ -th coefficient of transferred radiance  $(L'_p)_i$ ."). Sloan et al teaches generating a ray without accessing pre-calculated geometry associated with the object (5th paragraph of section 5: "In the first pass, for each  $p \in O$ , we cast shadow rays in the hemisphere about  $p$ 's normal  $N_p$ ..."), and "determining if a ray intersects a surface" (5th paragraph of section 5: "We tag each direction  $s_d$  with an occlusion bit,  $1 - V_p(s_d)$ , indicating whether  $s_d$  is in the hemisphere and intersects  $O$  again (i.e., is self-shadowed by  $O$ )."); however, Sloan et al does not teach ray tracing on a stream processor.

42. Purcell et al teaches "providing a stream processor capable of identifying a path associated with a ray" (page 705, 1<sup>st</sup> paragraph of section 2.3: "In order to guide the mapping of new applications to graphics architectures, we propose that we view next generation graphics hardware as a streaming processor."); 4<sup>th</sup> paragraph of section 3, titled "Streaming Ray Tracing": "The traversal kernel steps rays through the grid until the ray encounters a voxel containing triangles"); "generating a ray from a point on the object" (1<sup>st</sup> paragraph of section 3.1.2: "The traversal stage searches for voxels containing triangles...The second part steps along the ray enumerating those voxels pierced by the ray."); "determining if the path of the ray intersects a surface" (4<sup>th</sup> paragraph of section 3, titled "Streaming Ray Tracing": "The intersection kernel is responsible for testing ray intersections with all the triangles contained in the voxel.").

43. With regard to **claim 23**, Sloan et al teaches "a method for calculating an approximation to a transfer function defined by at least one basis function for rendering shading characteristics

of an object in real time, comprising: identifying a point on the object; calculating a lighting function for the point” (*1<sup>st</sup> paragraph of section 5: “As a preprocess, we perform a global illumination simulation over an object  $O$  using the SH basis over the infinite sphere as emitters.”; 6<sup>th</sup> paragraph of section 5: “For diffuse surfaces, at each point  $p \in O$  we further compute the transfer vector by SH-projecting  $M_p$  [transfer function] from [equation] (10).”*).

Sloan et al teaches “determining a direct illumination transfer function for the point” by means of a ray tracer (*1<sup>st</sup> paragraph of section 5: “As a preprocess, we perform a global illumination simulation over an object  $O$  using the SH basis over the infinite sphere as emitters.”; 3<sup>rd</sup> paragraph of section 5: “An initial pass simulates direct shadows from paths leaving  $L$  and reaching sample point  $p \in O$ .”*). Sloan et al teaches applying a ray tracing algorithm without accessing pre-calculated geometry associated with the object (*4<sup>th</sup> paragraph of section 4: “... $L_p(s)$  represents incident lighting assuming  $O$  was removed from the scene.”; 1<sup>st</sup> paragraph of section 6.2: “ $O$  itself should be removed from these renderings.”*). Sloan et al does not teach ray tracing on a stream processor in real-time.

44. Purcell et al teaches “applying a ray tracing algorithm through a stream processor” (*Figure 2, page 705*). Purcell et al teaches “determining a secondary lighting contribution in real time through a series of multiply and add operations applied to data resulting from the ray tracing algorithm” (*4<sup>th</sup> paragraph of section 3: “Additionally, the shading kernel may generate shadow or secondary rays; in this case, these new rays are passed back to the traversal stage.”*). With regard to the multiply and add operations for secondary lighting for secondary illumination, the triangle intersection stage (as shown in Figure 2 on page 705 of Purcell et al) occurs after the traversal stage. Figure 5 (Code for ray-triangle intersection) shows the dot product operation for

Art Unit: 2628

two vectors on lines 12, 14, and 15. Furthermore, Purcell et al refers to the method as “real-time ray tracing” in the first sentence of the 2<sup>nd</sup> paragraph of section 1.

45. With regard to **claims 6 and 23**, Sloan et al states that the “light gathering solution technique is a straightforward adaptation of existing approaches [7][25] and could be accelerated in many ways,” noting that [7] is a ray-tracing algorithm. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to accelerate the computation of radiance transfer as taught by Sloan et al, by using a stream processor to trace rays as taught by Purcell et al. The motivation for doing so would have been to achieve better performance tracing rays as stated by Purcell et al in section 2.3, which would be advantageous for interactive systems.

46. With regard to **claims 6 and 23**, Sloan et al admits that due to the limitations of contemporary consumer-level hardware (e.g. PIII-933 PC with an ATI Radeon 8500) the dot product of the captured samples and the basis functions must be performed in software. However, Sloan et al teaches, “ideally, this computation would be performed on graphics hardware” (3<sup>rd</sup> paragraph of section 6.2) and Sloan et al states, “we are interested in tracking fast-improving PC graphics hardware so that all computation, including transfer matrix transforms and SH-rotation, may eventually be performed on the GPU” (2<sup>nd</sup> paragraph of section 10). Therefore, at the time of the invention, it would have been obvious to a person of ordinary skill in the art to perform all computation on the GPU as suggested by Sloan et al. The motivation for doing so would have been to improve the performance of the method. Therefore, it would have been obvious to further modify the combination of Sloan et al and Purcell et al to obtain the invention as specified in **claims 6 and 23**.

Art Unit: 2628

47. **Claim 7** is met by the combination of Sloan et al and Purcell et al, wherein Purcell et al teaches a “stream processor capable of identifying a path associated with a ray includes, reading data associated with the ray” (4<sup>th</sup> paragraph of section 3: “The traversal kernel reads the stream of rays produced by the eye ray generator.”; 1<sup>st</sup> paragraph of section 3.1.2: “The second part steps along the ray enumerating those voxels pierced by the ray”); and “reading polygon data associated with the path” (4<sup>th</sup> paragraph of section 3: “The traversal kernel steps rays through the grid until the ray encounters a voxel containing triangles. The ray and voxel address are output and passed to the intersection kernel.”; Figure 4).

48. **Claim 8** is met by the combination of Sloan et al and Purcell et al, wherein Purcell et al teaches an “operation of generating a ray from a point on the object includes, determining a voxel traversed by a ray segment” (1<sup>st</sup> paragraph of section 3.1.2: “The traversal stage searches for voxels containing triangles...The second part steps along the ray enumerating those voxels pierced by the ray.”).

49. **Claim 9** is met by the combination of Sloan et al and Purcell et al, wherein Purcell et al teaches “if the path of the ray does not intersect the surface, then the method includes, determining a next voxel traversed by a next ray segment” (3<sup>rd</sup> paragraph of section 3: “If no hit occurs, the ray is passed back to the traversal kernel and the search for voxels containing triangles continues.”).

50. **Claim 10** is met by the combination of Sloan et al and Purcell et al, wherein Purcell et al teaches “if the ray intersects a surface, then the method includes, recording data associated with the location of the surface intersection” (4<sup>th</sup> paragraph of section 3: “If ray-triangle intersection (hit) occurs in that voxel, the ray and the triangle that is hit is output for shading.”); and

“generating a next ray” (*3<sup>rd</sup> paragraph of section 3.1.4: “The shading kernel optionally generates shadow, reflection, refraction, or randomly generated rays.”*); and “if the ray does not intersect a surface, then the method includes, reading data associated with a next voxel” (*3<sup>rd</sup> paragraph of section 3: “If no hit occurs, the ray is passed back to the traversal kernel and the search for voxels containing triangles continues.”*; *2<sup>nd</sup> paragraph of section 3.1.2: “After each step, the kernel queries the grid data structure which is stored as a 3D texture.”*); “advancing the ray through the next voxel” (*1<sup>st</sup> paragraph of section 3.1.2: “The traversal stage searches for voxels containing triangles...The second part steps along the ray enumerating those voxels pierced by the ray.”*); “and repeating if the path of the ray intersects the surface” (*1<sup>st</sup> paragraph of section 3.1.2: “The traversal stage searches for voxels containing triangles...The second part steps along the ray enumerating those voxels pierced by the ray.”*).

51. **Claim 11** is met by the combination of Sloan et al and Purcell et al, wherein Purcell et al teaches, “determining if the surface intersection is a closest surface intersection” (*Figure 3 (c)*).

52. **Claims 17-22** are rejected with the rationale of claims 6-11 respectively. Claims 17-22 are similar in scope to claims 6-11 recited as a computer readable medium with having program instructions.

53. With regard to **claim 24**, Sloan et al further teaches “the multiply and add operations are performed by a processor without calculating the lighting function at triangle corners” (*3<sup>rd</sup> paragraph of section 6: “The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader...Our pixel shader needs 8 instructions to perform the dot-product and stores LP’s coefficients in constant registers.”*). Sloan et al does not teach a stream processor. Purcell et al teaches a stream processor (*1<sup>st</sup> paragraph of section 2.3: “In*

Art Unit: 2628

*order to guide the mapping of new applications to graphics architectures, we propose that we view next-generation graphics hardware as a streaming processor."*

54. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to further modify the combination of Sloan et al and Purcell et al to incorporate computing the dot product in the stream processor. The motivation for doing so would have been to achieve better performance resulting from parallel computations, as suggested by Purcell et al in the third paragraph of section 2.3. Therefore, it would have been obvious to further modify Sloan et al with Purcell et al to obtain the invention specified in **claim 24**.

55. **Claim 26** is met by the combination of Sloan et al and Purcell et al, wherein Sloan et al teaches "repeating the identifying and the calculating for multiple points on the object" (*2<sup>nd</sup> paragraph of section 5: "An initial pass simulates direct shadows from paths leaving L and reaching sample points  $p \in O$ ...In each pass, energy is gathered to every sample point  $p$ ."*).

56. **Claims 33, 34, and 36-38** are rejected with the rationale of claims 23, 26, and 28-30 respectively. Claims 33, 34, and 36-38 are similar in scope to claims 23, 26, and 28-30 recited as a computer readable medium with having program instructions.

57. **Claims 16 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peter-Pike Sloan, Jan Kautz, John Snyder, "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments," July 2002, ACM Transactions on Graphics, Vol. 21 No. 3, p. 527-536 (Sloan et al) in view of Timothy J. Purcell, Ian Buck, William R. Mark, Pat Hanrahan, "Ray Tracing on Programmable Graphics Hardware," July 2002, ACM Transactions on Graphics, Vol. 21, No. 3, p. 703-712**

**(Purcell et al) and in further view of Robert L. Cook, "Stochastic Sampling in Computer Graphics," Jan. 1986, ACM Transactions on Graphics, Vol. 5, No. 1, p. 51-72 (Cook).**

58. As previously stated, the Sloan et al and Purcell et al combination meets the limitations of parent claim 12. With regard to **claim 16**, the Sloan et al and Purcell et al combination is silent as to biased or unbiased approximators. Cook et al teaches "applying one of a biased" approximator (*1<sup>st</sup> paragraph of section 5.2: "Sometimes we need to weight the samples...A better approach is importance sampling, in which the sample points are distributed so that the chance of a location being sampled is proportional to the value of the filter at that location."*; *3<sup>rd</sup> paragraph of section 5.2: "For example, for the reflection ray, we create a lookup table based on the specular reflection function."*) and "unbiased approximator" (*1<sup>st</sup> paragraph of section 5.1: "One way to distribute the rays in the additional dimensions is with uncorrelated random values."*).

59. As previously stated, the Sloan et al and Purcell et al combination meets the limitations of parent claim 23. With regard to **claim 25**, the Sloan et al and Purcell et al combination is silent as to biased or unbiased approximators. Cook et al teaches "applying one of a biased approximator" in the *1<sup>st</sup> paragraph of section 5.2 as shown in the paragraph above.*

60. Cook et al, Sloan et al and Purcell et al are analogous art because they are from the same problem solving area: computing illumination on graphics hardware. At the time of the invention it would have been obvious to a person of ordinary skill in the art to apply one of an unbiased approximator and the biased approximator as taught by Cook et al in the ray tracer taught by the Purcell et al and Sloan et al combination. The motivation for doing so would be to provide a pattern to distribute the rays, or in some cases, such as determining direct lighting, to use a biased approximator to direct where the rays will go to avoid unproductive computations (*Cook*

*et al* in the 1st paragraph of section 5.2 states that importance sampling “puts the samples where they will do the most good.”). Therefore, it would have been obvious to combine Sloan *et al* and Purcell *et al* with Cook to obtain the invention as specified in **claims 16 and 25**.

**61. Claims 27 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peter-Pike Sloan, Jan Kautz, John Snyder, "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments," July 2002, ACM Transactions on Graphics, Vol. 21 No. 3, p. 527-536 (Sloan *et al*) in view of Timothy J. Purcell, Ian Buck, William R. Mark, Pat Hanrahan, “Ray Tracing on Programmable Graphics Hardware,” July 2002, ACM Transactions on Graphics, Vol. 21, No. 3, p. 703-712 (Purcell *et al*) and in further view of U.S. Patent No. 6,268,860 to Bonello.**

62. As previously stated, the Sloan *et al* and Purcell *et al* combination meets the limitations of claim 31. With regard to claim 27, Sloan *et al* does not disclose delayed evaluation based on frames. Bonello discloses performing a calculation “for a portion of the multiple points during a first frame of image data, and performing the calculation for a remainder of the multiple points during a next frame of image data” (*lines 41-47 of column 8: " In this variation, the invention makes use of the knowledge that the illumination situation rarely changes between images (frames) that are adjacent in time, so it suffices to calculate precisely the relevance of the individual light sources for only every third frame, for example, and retain it in the display of the frames located between them in time."*).

63. Sloan *et al*, Purcell *et al* and Bonello are analogous art because they the same problem solving area: improving the efficiency of illumination calculations for image generation. At the time of the invention it would have been obvious to a person of ordinary skill in the art to defer



Art Unit: 2628

running illumination calculation on a portion of the data as taught by Bonello in the image generation system as taught by the Sloan et al and Purcell et al combination. The motivation for doing so would have been to reduce the amount of computation for displaying computer models for each frame as stated by Bonello in lines 31-32 of column 8, which would improve the performance of an interactive system.

64. Claim 35 is rejected with the rationale of claim 27. Claim 35 is claim 27 recited as a computer readable medium.

65. **Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Peter-Pike Sloan, Jan Kautz, John Snyder, "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments," July 2002, ACM Transactions on Graphics, Vol. 21 No. 3, p. 527-536 (Sloan et al) in view of Timothy J. Purcell, Ian Buck, William R. Mark, Pat Hanrahan, "Ray Tracing on Programmable Graphics Hardware," July 2002, ACM Transactions on Graphics, Vol. 21, No. 3, p. 703-712 (Purcell et al) and in further view of Tomas Möller, Eric Haines, "Real-Time Rendering," 1999, A.K. Peters, p. 68 (Möller et al).**

66. As previously stated, the Sloan et al and Purcell et al combination meets the limitations of claim 31. With regard to claim 32, Sloan et al does not disclose linear interpolation. Möller et al teaches, "linearly interpolating a color of the object across a polygon" (*2<sup>nd</sup> paragraph of p. 68: "In Gouraud shading [138], the lighting at each vertex of a triangle is determined, and these lighting samples (i.e., computed colors) are interpolated over the surface of the triangle."*).

67. Sloan et al, Purcell et al and Möller et al are analogous art because they are from the same problem solving area: real-time rendering. Möller et al states in the 3<sup>rd</sup> paragraph on page

Art Unit: 2628

68, "Most graphics hardware implement Gouraud shading because of its speed and much improved quality." At the time of the invention it would have been obvious to a person of ordinary skill in the art to incorporate interpolation as taught by Möller et al in the color computations of the combination of Sloan et al and Purcell et al. The motivation for doing so would have been to give a "smooth look to curved surfaces" (3<sup>rd</sup> paragraph of page 68, Möller et al) without having to compute the color attribute at each point on the surface of the object.

**68. Claims 40, 41 and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jan Kautz, Peter-Pike Sloan and John Snyder, "Fast, Arbitrary BRDF Shading for Low-Frequency Lighting Using Spherical Harmonics," June 26, 2002, Proceedings of the 13th Eurographics Workshop on Rendering, p. 291-296, 335 (Kautz et al) in view of U.S. Patent No. 6,639,595 to Drebin et al.**

69. With regard to **claims 40, 41 and 45**, Kautz et al does not expressly disclose "a display and video console" and "linear interpolation on the rendered object." With regard to claims 40 and 41, Drebin et al discloses "a computing device, comprising: a graphics processing unit (GPU) capable of determining lighting characteristics for an object in real time" (*lines 58-60 of column 4: "In this example, system 50 is capable of processing, interactively in real time, a digital representation or model of a three-dimensional world."*; *lines 9-13 of column 9: "As discussed above, transform unit 300 in the example embodiment performs lighting in addition to geometric transformations, clipping, culling and other functions. In the example embodiment, transform unit 300 supports lighting in hardware as a per-vertex calculation."*) "a display screen in communication with the GPU, the display screen configured to present image data representing the object" (*lines 3-5 of column 6: "The graphics and audio processor 114*

Art Unit: 2628

*processes these commands to generate interesting visual images on display 59...”) a computing device that is a “video game console” (lines 5-8 of column 5: “To play a video game or other application using system 50, the user first connects a main unit 54 to his or her color television set 56 or other display device by connecting a cable 58 between the two.”).*

70. With regard to **claim 45**, Drebin et al discloses “the GPU is further configured to render the object through a process involving linear interpolation, such that the lighting characteristics are applied to the rendered object” (*lines 11-16 of column 9: “In the example embodiment, transform unit 300 supports lighting in hardware as a per-vertex calculation. This means that a color (RGB) value can be computed for every lit vertex, and that these colors can be linearly interpolated over the surface of each lit triangle.”*).

71. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to incorporate the lighting and shading computations disclosed by Kautz et al in the video game console attached to a display device as disclosed by Drebin et al. The motivation for doing so would have been to allow a user to interact with the virtual models. Therefore, it would have been obvious to combine Kautz et al with Drebin et al to obtain the invention specified in **claims 40, 41 and 45**.

72. **Claims 43 and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jan Kautz, Peter-Pike Sloan and John Snyder, “Fast, Arbitrary BRDF Shading for Low-Frequency Lighting Using Spherical Harmonics,” June 26, 2002, Proceedings of the 13th Eurographics Workshop on Rendering, p. 291-296, 335 (Kautz et al) in view of Timothy J. Purcell, Ian Buck, William R. Mark, Pat Hanrahan, “Ray Tracing on Programmable**

**Graphics Hardware,” July 2002, ACM Transactions on Graphics, Vol. 21, No. 3, p. 703-712 (Purcell et al).**

73. With regard to **claims 43 and 44**, Kautz et al discloses an optional step of precomputing radiance transfer using a global illumination algorithm in the first paragraph of section 5:

“Transfer matrices are derived using an offline global illumination simulation but can be applied at run-time to arbitrary low-frequency lighting.” Kautz et al does not expressly disclose a “the code that is replicated multiple times is configured to process one of a ray tracing algorithm and multiply and add operations for data derived from the ray tracing algorithm.”

74. With regard to **claim 43**, Purcell et al discloses “the code that is replicated multiple times is configured to process one of a ray tracing algorithm (*Figure 2: A streaming ray tracer; 5<sup>th</sup> paragraph of section 6: “We implement ray tracing kernels as fragment programs.”*) and multiply and add operations for data derived from the ray tracing algorithm” (*Figure 5: code for ray tracing triangles.*). Purcell et al does not use the explicit language “multiply and add operations”, but one of ordinary skill in the art would recognize that this feature is inherent from lines 12, 14, and 15 in the code presented in Figure 5, where the dot product of two vectors is computed.

75. With regard to **claim 43**, at the time of the invention, it would have been obvious to a person of ordinary skill in the art to use the ray tracing method disclosed by Purcell et al as the global illumination method in the system and method disclosed by Kautz et al. The motivation for doing so would have been to “obtain self-shadowing and self-scattering effects on anisotropic materials” (see section 5 of Kautz et al).

Art Unit: 2628

76. With regard to **claim 44**, Purcell et al discloses “the ray tracing algorithm determines a direct illumination lighting characteristic (*4<sup>th</sup> paragraph of section 3: “If ray-triangle intersection (hit) occurs in that voxel, the ray and the triangle that is hit is output for shading”*) and the multiply an add operation determine a secondary lighting characteristic (*4<sup>th</sup> paragraph of section 3: “Additionally, the shading kernel may generate shadow or secondary rays; in this case, these new rays are passed back to the traversal stage.”; Figure 5*).

77. With regard to **claims 43 and 44**, Purcell et al discloses the method approaches “real-time,” but does not disclose the method’s performance is in “real-time.” At the time of the invention, it would have been obvious to a person of ordinary skill in the art to perform the ray-tracing method disclosed by Purcell et al in real time. The advantage of incorporating a real-time global illumination algorithm in Kautz et al would have been to eliminate the need to compute the transfer matrices offline (*1<sup>st</sup> paragraph of section 5*) to permit a user to interact with the rendered image. Therefore, it would have been obvious to combine Kautz et al with Purcell et al to obtain the invention specified in **claims 43 and 44**.

### *Conclusion*

78. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Nathan A. Carr, Jesse D. Hall and John C. Hart, "The Ray Engine," March 2002, Tech. Rep. UIUCDCS-R-2002-2269, Department of Computer Science, University of Illinois, teach a ray-triangle intersection computation using a pixel shader. John D. Owens, William J. Dally, Ujval J. Kapasi, Scott Rixner, Peter Mattson, Ben Mowery, “Polygon Rendering on a Stream Architecture,” August 2000, Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware, p.23-32, teaches a stream processor. Stephen H. Westin,

Art Unit: 2628

James R. Arvo, Kenneth E. Torrance, "Predicting Reflectance Functions from Complex Surfaces," July 1992, Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, p.255-264 teaches using a spherical harmonic basis for lighting calculations.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason M. Repko whose telephone number is 571-272-8624. The examiner can normally be reached on Monday through Friday 8:30 am -5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on 571-272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JMR

  
ULKA CHAUHAN  
SUPERVISORY PATENT EXAMINER